VOL. 5 NO. 6
JUNE 1983

neat notes

THE NEWSLETTER OF NEW ENGLAND APPLE TREE

June 1983 N.E.A.T. Meeting

The June meeting will be held at 7:30 P.M. on the third Wednesday of the month (as usual), June 15, 1983 at the Mitre cafeteria. Mitre is on rte 62 in Bedford Ma. Take route 3 north from rte 128 to the first exit (rte 62).

JUNE'S MEETING

Our speaker for June is Dave Szetela, Technical Editor for NIBBLE magazine and Product Development Manager for Micro-SPARC. Dave will present the 'ins and outs' of commercial software development, with special advice for customer support, advertising, and budgeting. See the accompanying article on Page 5.

AN EXCITING SCHEDULE IS SHAPING UP FOR THE SUMMER!
THE FUTURE SCHEDULE IS TENTATIVELY AS FOLLOWS:

July 20,1983----------LISA, Jim Miller

August 17,1983--------68000 in an Apple,
Joe Stivaletta

September 21,1983-----CompuServ

October 19,1983-------

November 16,1983------

December 21,1983------

NEAT NOTES -New England Apple Tree- Box 2652, Woburn, Ma 01801

APPLE®..MEET THE OPERATOR™

MODEM TECHNOLOGY WITHOUT PAYING THE PRICE ON THE BOX

If you're sold on telecomputing but hung up on the price, check out THE OPERATOR. For just $159.95 *complete,* you'll find no other modem that gives you crisper Apple communications.

And you don't have to be a technical wizard to get started. THE OPERATOR includes all the sophisticated technology on a single circuit board. No external box. No need for expensive add-ons. THE OPERATOR is ready to plug into any communications system, world-wide, the day she arrives.

This modem comes with complete documentation even a beginner can understand, *plus* start-up software programs (on disk), so you don't have to worry about added expenses or frustrating delays.

$159.95 COMPLETE

NO EXTERNAL BOX
HARDWARE • DOCUMENTATION
START-UP PROGRAMS (ON DISK)

FEATURES

110/300 baud • Full & Half Duplex
Auto Answer w/Disconnect
Rotary, Touch-Tone,™ Key Set Compatible
Totally Self-Contained (No RS-232 Required)
Single Modem Chip Reliability
Interfaces with Most Communications Packages

SPECIAL INTRODUCTORY OFFER

Order THE OPERATOR today and be our guest for 60 minutes on T•I•M•E•C•O•R's international communications network via GTE Telenet.

YES, I need THE OPERATOR™ Rush her to me today!

Name _____

Company _____

Address _____

City _____ State ____ ZIP _____

Phone (____) _____ [ ] Dealer [ ] Business [ ] Home

Computer _____

☐ Check enclosed ($159.95*) ☐ Send more information

*Mass. residents add 5% sales tax. Shipping & handling included for continental U.S.

T•I•M•E•C•O•R
Four Longfellow Place, P.O. Box 8928, Boston, MA 02114

FOR FASTER SERVICE CALL TOLL FREE

MasterCard and VISA only

1-800-852-5000

THE OPERATOR is a trademark of T•I•M•E•C•O•R.
Apple is a registered trademark of Apple Computer, Inc.
Touch-Tone is a trademark of American Telephone & Telegraph.

T·I·M·E·C·O·R

REACHING PEOPLE THROUGH TECHNOLOGY

The International Modem Exchange Corporation

Four Longfellow Place, P.O. Box 8928, Boston, MA 02114 VOICE: (617) 720-4090 MODEM: (617) 720-3600

neat notes

THE NEWSLETTER OF NEW ENGLAND APPLE TREE

NEAT NOTES STAFF

Editor/Publisher.....
John Shaw,643-2679
Editor/Writer-Pascal...
Jerry Connelly(603)895-4090
Editor/Writer-Assembly Language
John Shaw,643-2679
Games Reviewer.....
Mark Hessman,475-2617
Advertising Manager....
HELP !!!

N.E.A.T. OFFICERS and CHAIRMEN

President..Jerry Connelly,497-9050
Vice-pres.....Dave Reed,692-7547
Vice-pres.....Lou Tarsa,823-8106
Secretary.Jeannine Giffee,497-9050
Treasurer...Larry Starnes,263-1795
Library.....
Input.....M.Gooding,264-4237
Sales.....D.McMullin,444-4304
DOM (at D.Goldstein,273-0567
meeting).E.Mac Lauchlan,878-2229
Outside User Group Contact.....
Larry Starnes,263-1795
Membership.Joe Stivaletta,861-1899
Auditor.....Don Foster
Greeter.....Jack Thompson,253-8608

SPECIAL INTEREST GROUPS

SIGED.....Jack Sheehan,924-1197
SIGCP/M....Joe Stivaletta,861-1899
SIGASSY.....John Shaw,643-2679
SIGCOMM.....
SIGPASCAL..Jerry Connelly,497-9050
SIGNEW.....Lou Tarsa,547-3289

Copyright 1982, New England Apple Tree, All rights reserved.

JUNE HIGHLIGHTS

Calendar
by Raines Cohen.....Page 4
Catalog With Class
by John J. Shaw.....Page 4
JUNE SPEAKER: Dave Szetela
.....Page 5
Comdex/Spring, Atlanta GA.
by Dave Reed.....Page 6
Machine Language Spoken Here
by Dr. J. .....Page 8
Boston Applefest: A Review
.....Page 12
Disk Of The Month
.....Page 14
Membership Application and
DOM Order Blank.....Page 15

CALENDAR
Compiled by Raines Cohen

June 25 2:30PM
Newton Apple Kids
Raines Cohen's House
call 965-0033

July 6 7:00PM
New England Computer Society
Mitre Cafeteria
call 646-3080

July 9 10:00AM
North Shore Apple Group
Computer City, Danvers
call 546-3104

July 16 9:30AM
Apple Guild
Massasoit Community College
call 767-1303

July 20 7:30PM
New England Apple Tree
Mitre Cafeteria
call 497-9050

July 23 2:30PM
Newton Apple Kids
Raines Cohen's House
call 965-0033

CATALOG WITH CLASS

By John J. Shaw

Have you ever admired the way VISICALC (and the other VISI-programs) gives you a CATALOG of your disk? VISICALC's CATALOG routine displays one TEXT file at a time and lets you either select the file, or continue the search. Here's a neat little disk utility you can add to your own programs which mimics VISICALC's routine. This subroutine gives a catalog of your disk by showing you one text file at a time, and lets you either select the file shown, continue the search, or abandon the catalog altogether. The selected file name is returned in the string variable N$.

I've been using this routine in programs which ask for text files to read/write and find it beats having to type the name of the files directly.

```
15000 CT = 224: POKE CT,32: POKE
    CT + 1,227: POKE CT + 2,3: POKE
    CT + 3,76: POKE CT + 4,217: POKE
    CT + 5,3
15010 TRK = 17:RW = 1: POKE 47083
    ,0: POKE 47084,TRK: POKE 470
    91,0: POKE 47092,RW: POKE 47
    082, PEEK ( - 21912)
15020 LOC = 38400: POKE 47088,LOC
     - INT (LOC / 256) * 256: POKE
    47089, INT (LOC / 256)
15030  HOME : VTAB 13: PRINT "PRE
    SS ";: INVERSE : PRINT "S";:
     NORMAL : PRINT " TO SELECT
    A FILE": PRINT : PRINT "PRES
    S <RETURN> TO CONTINUE THE S
    EARCH": PRINT : PRINT "PRESS
     <ESC> TO STOP THE SEARCH": VTAB
    20
15040  FOR SEC = 15 TO 1 STEP  -
    1: POKE 47085,SE: CALL CT: FOR
    X = LOC + 11 TO LOC + 221 STEP
    35: IF  PEEK (X) = 0 THEN SE
    C = 0:X = LOC + 255: GOTO 16
    000
15050  IF  PEEK (X) = 255 THEN 16
    000
15060 P =  PEEK (X + 2):P = P - 1
    28 * (P > 127): IF P > 0 THEN
    16000
15070 N$ = "":Y1 = 0:Y2 = 0: FOR
    Y = X + 3 TO X + 32:N$ = N$ +
     CHR$ ( PEEK (Y) - 128):Y1 =
    Y1 + 1: IF  PEEK (Y) <  > 16
    0 THEN Y2 = Y1
15075  NEXT Y:N$ =  LEFT$ (N$,Y2)

15080  PRINT N$;"  ";: GET A$: PRINT
    : VTAB 20: CALL  - 868
15090  IF A$ = "S" THEN 16020
15095  IF A$ =  CHR$ (27) THEN N$
     = "": GOTO 16020
16000  NEXT X,SEC
16010 N$ = ""
16020  HOME : RETURN
```

JUNE SPEAKER: DAVE SZETELA

Quick! What do NIBBLE magazine, Dave Szetela, Alan and Valerie Floeter, Call-A.P.P.L.E., THE ASSEMBLER/MACROSOFT, June's program, and YOU have in common? Give up? Consider the following:

1) NIBBLE magazine's subsidiary, Micro-SPARC, is quickly becoming one of the fastest growing software producers in New England.

2) Dave Szetela is the Technical Editor of NIBBLE, and Product Development Manager of Micro-SPARC.

3) Alan and Valerie Floeter have been contributing editors to NIBBLE for several years now, and have just recently had two new programs: THE ASSEMBLER and MACROSOFT published through Micro-SPARC.

4) Alan and Valerie wrote the lead article to the May 1983 Call-A.P.P.L.E describing how to market your own 'million dollar' program.

5) Dave Szetela will be our program speaker for June; HIS talk will also deal with the ins and outs of marketing your own 'million dollar' program. If we're lucky, we will also get him to tell us something about THE ASSEMBLER and MACROSOFT.

6) You will 'close the loop' (so to speak) by getting the inspiration to go out into this brave world and market that program you have been working on 6 hours a night for the last four months!

Whew! That almost sounds like a modern day 'PEYTON PLACE'! The current events may not be as racy as the aforementioned book, but the players involved are laughing all the way to the bank. And why shouldn't they? New England, not 'Silicon Valley' or Bellevue Washington, is the home of the micro computer software industry! If you don't believe me, take a look around at some of the area's BIG players (Lotus, Spinnaker Software, Software Arts, Infocom, Micro-SPARC, Anthro-Digital, Software Technology, Microcom) and some of the new players on the scene (Synoptic Software, Wadsworth).

For the last several years, NIBBLE magazine has supported the would-be software author with a series of articles describing copyright laws, software protection, software marketing, and even how to develop a realistic budget for your fledgling firm. This industry has recently attracted a swarm of 'expert' gadflys who, for a $600 1-day seminar, will gladly sell a 'sky's the limit' future where you can make a quick million or two if you can tell the difference between a ROM chip and a REM statement. NIBBLE remains one of the few voices reminding us that a company's revenues comes from hard work, budgeting and planning, customer and sales support, and above all a competitive product.

If you could not attend the June meeting, but would like to learn what Dave had to say, I suggest you write him directly:

Dave Szetela
Technical Editor
NIBBLE
P.O. Box 325
Lincoln, MA 01773

COMDEX/SPRING, ATLANTA GA.
by Dave Reed

This year Atlanta, Ga., home of the "good old boys" played host to over 700 exhibitors who spent the better part of the week convincing dealers to carry their products. Well, Atlanta hasn't changed much since I was last there four years ago. Oh yes Virginia, there are more new buildings in the city and a lot more industry in the suburbs and some of the "good old boys" have traded in their pickups for vans but the city is still the quite relaxed center of the great American Southland that I remembered and enjoyed.

CONFERENCE SESSIONS

The COMEX Conference offered 56 sessions designed to focus in on business, marketing and financial subjects of special relavance to Independent Sales Organizations (ISO's). It was obviously impossible to cover all of the sessions but they were professionally taped and you had the opportunity to buy good quality tapes of any and all sessions. I attended sessions on Desktop Networking and Desktop Database Advances which were good and one on Graphics Marketplace Forecast which was terrible. I have access to tapes on the first two and twenty-two pages of notes so I should be able to recap them for you in the near future. I also have tapes on two Marketing Knowhow, Introduction sessions: Launching That Proprietary Product and Direct-Response Workshop which were both very good.

I also sat in on the beginning of a very interesting panel session entitled ISO/Vendor Forum. This panel was made up of a wonderfully diverse group including the very vocal Portia Isaacson, a well know market researcher and writer from Richardson, Tx. and our own quiet but equally well known Dan Bricklin, co-author of VisiCalc and Chairman of the Board and Executive V.P., Software Arts, Inc. It also included Rick Inatome, Pres. Computer Mart, Inc., Chuck Peddle, Pres. Victor Technology, Inc., Richard Connaughton, Group Director ISO, Wang Labs. and was chaired by Sandy Sansing, Pres. Digital Systems of Florida, Inc. I only wish I had had the time to stay for the full session.

EXHIBITS

Because the COMDEX is aimed at the Independent sales organizations, manufacturers like IBM and Apple were not there pushing their personal computers (IBM did have a booth for other products and there were PC's on display). The newer and smaller computer manufacturers were there with portables and PC and Apple lookalikes (I can't for the life of me understand why they think there is room for more "me too" products).

The portables by far dominated the new computer catagory with entrys from Tandy, Epson, NEC, Compaq, Grid Systems and many others. The most interesting however was from a new startup in Campbell, Ca., Gavilan Computer Corp. They offer a nicely packaged sixteen bit micro with LCD display and built in 3 inch micro floppy drive with add on RAM or ROM cartridge and printer. All this is battery operated and smaller than your briefcase. Sorry, it is not Apple compatible.

All in all there were 77 vendors listed under Computers-Micros, 8 under -Mini,

32 under -personal, 15 under -portable and 61 under -Small Business. We therefore see an almost 2 to 1 focus on business over personal computers.

There was certainly a large amount of space dedicated to the marketing of board sets, peripherals, and communications equipment and although most of it addressed the IBM PC a good amount addressed the Apple as well. In this area the Mice (or is it Mouses) were one of the newest. They were being demonstrated attached to the Apple running a word processing program and the mouse was functioning as cursor control, scroll, and left and right arrow. In this example the mouse was interfaced using a serial port and was being read alternately with the keyboard and as the position, and the button depressions were sensed from the mouse, character strings were inserted into the keyboard buffer for the program to interpret in its usual way. Thus the application did not have to be changed only the I/O routines.

There were of course many floppy and hard disk drives and cassette drives all of which require some kind of new interface and power supply but what caught my eye was a new product from Amlyn Corp. Amlyn has been making a device for a while which contains a removable cartridge which holds five floppy diskettes. When the software tells the controller to select another drive the Amlyn device replaces the current diskette into the cartridge and draws the selected one out onto the spindle and loads the head. This selection process is slower than selecting your second drive but it offers on line access to five diskettes instead of one and leaves the other drive for reading Apple compatible software in a two drive system. Up until now this device has been compatible only with the higher transfer rates of eight inch drives but now there is a version which runs at a slower speed and is compatible with mini-floppy controlers. Each floppy is recorded at 170 tracks to the inch and offers 160 tracks per diskette for a capacity of 1Mbyte per diskette or 5Mbytes per cartridge. Now your controller still has to be capable of addressing six drives (and the Apple is not) so don't run out and buy one tomorrow but it offers interesting possibilities for the future.

On the Software side there were numerous products in all catagories: games, education, word processing, database management, spreadsheet, accounting, communications and networking, utility and operating systems. The one thing that most of the new ones have in common is an easier to use operator interface and to provide this they generally require more storage in memory and on disk and faster disks (read Winchester here) and more processing power (read 16 bits here).

The newest and most interesting were in a group which I will call Application Management Systems (AMS). These are programs which run between the operating systems and the application programs. They allow the user to easily move from application to application and pass data from one program to another. MicroPro has one called Starburst but the demo which I found most impressive was of Quarterdeck. This was operated on an IBM PC with a hard disk and a mouse and presented a multi-window display on the screen much like the Apple Lisa.

One window showed the menu associated with the AMS itself and the other two were the normal application program menus. The demo read in data using Dow Jones, manipulated the numbers using SuperCalc, put the results in a report under WordStar, and filed the data away using a Data Base Management System.

There will be some interesting products on the shelves as a result of this show but I am sure we Apple affectionados will find more of direct interest at the upcoming AppleFest.

MACHINE LANGUAGE SPOKEN HERE

By Dr. J

The doctor is in.

Hello gang. Originally, I planned to show you how to add two floating point arrays from machine language. However, I decided to push that topic back to next month's column, and address instead a question two different people asked me at APPLEFEST: 'how can I tell Applesoft the location of my machine language routine(s) after I load them into memory?'

A very good question. You see, up to now we have had to call our machine language routines using a 'CALL' statement followed by the numeric location of the routine's "entry point", such as: CALL 768 (if our routine begins at decimal location 768) or CALL 37888 (if our routine begins at decimal location 37888). What do we do if we forget the starting address for a machine language routine? We're stuck, right?

Fortunately, we can design our machine language routines to tell us where their entry points are located in memory. Moreover, we will be able to call these routines by NAME!

Suppose we had a machine language routine called ARITHMETIC.OBJ0 that actually contained four different subroutines: ADD, SUBTRACT, MULTIPLY, and DIVIDE. Each subroutine expects an argument list having three floating point variables; the particular arithmetic operation is performed on the first two variables in the list and the result is stored in the third variable. Now, it would be nice if we could call these routines by name from Applesoft in the following manner:

```
100 A = 30
120 B = 60
140 C = 0
160 CALL ADD,A,B,C
180 PRINT C
200 CALL SUB,A,B,C
220 PRINT C
240 CALL MULT,A,B,C
260 PRINT C
280 CALL DIV,A,B,C
300 PRINT C
```

Wouldn't you agree that the statement

```
CALL ADD,A,B,C
```

tells you more about the machine language routine you are using than the statement:

```
CALL 97847,A,B,C
```

The trick here is that most assemblers will let us assign labels to particular points in our machine language code. During assembly, the assembler will determine that a particular point is located at some memory location and assign the (2-byte) value of that location to its corresponding label. For example, in the short machine program:

```
        ORG $300
START   LDA #$01
        STA $9600
```

the label START will be assigned the hexadecimal value $300! Can our machine code get at this value itself? Sure it can! Again, most assemblers support immediate addressing wherein you can load the accumulator, Y-register, or X-register with the high/low order bytes of the address of a label. The DOS TOOLKIT assembler will let you load the low order byte of an arbitrary label, LABEL, using the commands:

```
LDA #>LABEL,
LDY #>LABEL,  or
LDX #>LABEL
```

I believe the MERLIN and LISA assemblers replace #>LABEL with #LABEL. The DOS TOOLKIT assembler will let you load the high order byte of an arbitrary label using the commands:

```
LDA #<LABEL,
LDY #<LABEL,  or
LDX #<LABEL
```

I believe the MERLIN assembler replaces #<LABEL with #>LABEL, and the LISA assembler replaces #<LABEL with /LABEL.

What good does any of this do us? Well, the Applesoft ROM routine GIVAYF will take any 2-byte number stored in the accumulator (high byte) and Y-register (low byte), convert it into a floating point number, and save the result in the floating point accumulator, FAC. We can then assign that floating point value to any Applesoft variable we choose. For example, in the short program:

```
        ORG $300
START   LDA #$01
        STA $9600
        LDA #<START
        LDY #>START
        JSR GIVAYF
```

the address to START will be loaded into FAC; from there we can assign this address to an Applesoft variable and establish a link between our Applesoft program and our machine language routine.

Getting back to our machine language program ARITHMETIC.OBJ0, we need to add two 'front end' routines to our main routine. The first routine will reset the Apple's Ampersand, '&', vector and have it point to the second routine. That is, we will be able to call the second routine from Applesoft without actually having to know where it begins. The second routine is called from Applesoft with the command:

```
& ADD,SUB,MULT,DIV
```

and assigns the appropriate entry points to the variables following the ampersand. The individual machine language sub-routines can then be called from the following program:

```
60  D$ = CHR$(4)
70  REM RUN THE FIRST
        'FRONT END'
        ROUTINE IN
        ARITHMETIC.OBJ0
        TO RESET THE
        '&' VECTOR.
80  PRINT D$"BRUN
    ARITHMETIC.OBJ0"
90  REM THE & VECTOR NOW
        POINTS TO THE
        SECOND 'FRONT END'
        ROUTINE
100 A = 30
120 B = 60
140 C = 0
145 REM NOW CALL THE
        SECOND 'FRONT
        END' ROUTINE
        TO ASSIGN THE
        MACHINE LANGUAGE
        ENTRY POINTS TO
        THE VARIABLES
        ADD, SUB, MULT,
        AND DIV.
150 & ADD,SUB,MULT,DIV
155 REM WE ARE NOW READY
        TO CALL THE
        INDIVIDUAL MACHINE
        LANGUAGE SUBROUTINES
        BY NAME.
160 CALL ADD,A,B,C
180 PRINT C
200 CALL SUB,A,B,C
220 PRINT C
240 CALL MULT,A,B,C
260 PRINT C
280 CALL DIV,A,B,C
300 PRINT C
320 CALL ADD,B,C,A
340 PRINT A
```

The machine language routine ARITHMETIC.OBJ0 accomplishes all of this and is shown in Listing 1. The first 'front end' routine is located at lines 32 thru 45 and the second front end routine (labeled START) is located at lines 46 thru 85. During assembly, the label START is assigned the 2-byte value denoting the start of the second front end routine. The first front end routine simply loads these two bytes into the accumulator (lines 36 and 38) and stores them at hexadecimal locations $3F6 (low byte of the address) and $3F7 (high byte of the address). This routine also stores the op-code for a machine language 'JMP' command at memory location $3F5. Now when Applesoft comes across the ampersand symbol, &, it will find an instruction to JuMP to the start of the second front end routine.

The second front end routine takes the machine language labels ADD, SUB, MULT, and DIV, and converts their 2-byte values into floating point values (through subroutine GIVAYF). These floating point values are then assigned to the appropriate variables appearing in the parameter list when this routine is called from Applesoft (e.g., line 150 in the short program above). For example, the numeric value of the high order address byte for the routine ADD, $94, is stored in the accumulator and the value of the low order address byte, $52, is stored in the Y-register. Subroutine GIVAYF takes this 2-byte pair ($9452), converts it into a floating point number and saves the result in the floating point accumulator, FAC. This value is then assigned to the first variable appearing in the aforementioned parameter list. [A side note: the hexadecimal number $9452 normally translates into the floating point value 37970. However, the subroutine GIVAYF translates this into the signed floating point value -27566. Fortunately, this creates no problems because a CALL 37970 statement from Applesoft goes to the same machine language location as a CALL -27566 statement].

The real 'guts' of the machine language routine is found in lines 86 thru 133. This routine is actually made up of the four separate subroutines ADD, SUB, MULT, and DIV. Each routine reads the variables passed to it from Applesoft, performs an arithmetic operation on the first two parameters, and returns the result in the third. As indicated in lines 160 thru 320 in the Applesoft program above, these four subroutines will accept any three variables (compare line 160 with line 320).

This is, in nutshell, how to go about telling Applesoft where to find the starting locations to your machine language routines. The beauty of this method is it allows you to acquire the entry points to several subroutines within a larger machine language routine, and allows you to call these subroutines by name! Moreover, the second front end routine in Listing 1 doesn't care what variable names you give it, just as long as you give it as many variables as there are labels to assign. This being the case, we could have just as easily replaced the variable names ADD, SUB, MULT, and DIV with the variable names E1, E2, E3, and E4 in line 150 of the Applesoft program above.

Well, that about wraps it up for now. I hope you have fun playing with this new method. If you feel ambitious, you may want to re-write some our earlier machine language routines to use this new method. Next month, we'll return to our foray into the use of floating point arrays from machine language. Also, look for a review of a new assembler (called, simply enough, THE ASSEMBLER) from MicroSPARC, the publishers of NIBBLE magazine!

Till then, good luck!

The doctor is out.

```
                      ; THIS ROUTINE WILL ALLOW US
                      ; TO ASSIGN SPECIFIC NAMES TO
                      ; OUR MACHINE LANGUAGE ROUTINES,
                      ; AND THEN CALL THESE ROUTINES
                      ; BY NAME FROM APPLESOFT.
                      ;
                      ; THE INSTRUCTIONS ARE PROVIDED
                      ; IN THE ACCOMPANYING ARTICLE
                      ;
                      ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                      ;
                      ; SET UP POINTERS
                      ;
DFE3:                 PTRGET  EQU  $DFE3    ; GET VARIABLE POINTER
DEBE:                 CHKCOM  EQU  $DEBE    ; CHECK FOR COMMA
EAF9:                 MOVFM   EQU  $EAF9    ; MOVE (Y,A) INTO FAC
EB2B:                 MOVMF   EQU  $EB2B    ; MOVE FAC INTO (Y,X)
E2F2:                 GIVAYF  EQU  $E2F2    ; FLOAT INTEGER IN A,Y
E7BE:                 FADD    EQU  $E7BE    ; ADD FAC TO THE NUMBER
0000:                 ; POINTED TO BY (Y,A) AND KEEP IN FAC
E97F:                 FMULT   EQU  $E97F    ; DITTO FOR MULT.
EA66:                 FDIV    EQU  $EA66    ; DITTO FOR DIVISION
E7A7:                 FSUB    EQU  $E7A7    ; DITTO FOR SUBTRACTION
0000:                 ;
0000:                 ; BEGIN THE PROGRAM
0000:                 ;
0000:                 ----- NEXT OBJECT FILE NAME IS ARITHMETIC.OBJ0
9400:                       31          ORG  $9400
9400:                       32 ;
9400:                       33 ; FIRST, SET UP THE AMPERSAND
9400:                       34 ; '&' POINTER BY LOADING A 'JMP' INSTRUCTION
9400:                       35 ; AT MEMORY LOCATION $3F5.
9400:A9 10                  36          LDA  #)START  ; GET THE LOW BYTE OF
9402:8D F6 03               37          STA  $3F6     ; THE STARTING ADDRESS
9405:A9 94                  38          LDA  #(START  ; GET THE HIGH BYTE
9407:8D F7 03               39          STA  $3F7
940A:A9 4C                  40          LDA  #$4C     ; OP-CODE FOR 'JMP'
940C:8D F5 03               41          STA  $3F5     ; STORE IT
940F:                       42 ;
940F:                       43 ; FINISHED, RETURN TO BASIC
940F:                       44 ;
940F:60                     45          RTS           ; RETURN TO BASIC
```

BOSTON APPLEFEST: A REVIEW

In our review of last year's, APPLEFEST, we left my buddy Elliot buried under a two foot stack of promotional material from the show. Well, things haven't changed much; Elliot is still buried under a ton of promotional material, only now it stretches another two feet towards his ceiling and includes five boxes of back issues to APPLE ORCHARD magazine he picked up at 5:30 on the closing day of the show. The town of Lexington has declared him a fire hazard, although my guess is they'll first have to find him inside all that rubble before they can serve him the notice.

Before we begin the review of this year's APPLEFEST, El and I would like to thank all the individuals who lent NEAT a hand. All told, we had 20 volunteers who helped to spread **THE WORD** at the NEAT booth. We would like to extend our appreciation to:

Gary Widiger
David Neumann
Bob & Carol Morris
David Lynn
Roger Geffen
Alonso Garza
Terry Scheller
Mat Leupold
Tom Hamilton
Joe Stivaletta
Guy Desloovres
Raines Cohen
Peter Fishman
Sam Ezust
Mark Hessman
Rob Josephs
Alan Kleinman
Eileen Entin
Tavit Smith

El and I don't Know what you said

```
                        LDA #KMULT   ; REPEAT FOR THE
                ; 'MULT' ROUTINE
                        LDY #>MULT
                        JSR GIVAYF
                        JSR PTRGET
                        TAX
                        JSR MOVMF
                        JSR CHKCOM
                        LDA #KDIV    ; REPEAT FOR THE
                ; 'DIV' ROUTINE
                        LDY #>DIV
                        JSR GIVAYF
                        JSR PTRGET
                        TAX
                        JSR MOVMF
                        RTS          ; RETURN TO BASIC
                ;
                ; FINALLY, WE CAN BEGIN THE REAL PROGRAM:
                ; THE SUBROUTINES ADD, MULT, DIV, AND SUB
                ;
9452:20 BE DE   ADD     JSR CHKCOM   ; REMOVE FIRST COMMA
9455:20 E3 DF           JSR PTRGET   ; GET POINTER TO VARIABLE 'A'
9458:20 F9 EA           JSR MOVFM    ; MOVE ITS CONTENTS INTO FAC
945B:20 BE DE           JSR CHKCOM   ; REMOVE SECOND COMMA
945E:20 E3 DF           JSR PTRGET   ; GET POINTER TO VARIABLE 'B'
9461:20 BE E7           JSR FADD     ; FAC = A + B
9464:20 BE DE           JSR CHKCOM   ; REMOVE THIRD & LAST COMMA
9467:20 E3 DF           JSR PTRGET   ; GET POINTER TO VARIABLE 'C'
                        TAX          ; TRANSFER ACC TO X-REG.
946A:AA
946B:20 EB      SUB     JSR MOVMF    ; C = FAC = A + B
946E:60                 RTS          ; RETURN TO BASIC
946F:20 BE DE           JSR CHKCOM   ; REMOVE FIRST COMMA
9472:20 E3 DF           JSR PTRGET   ; GET POINTER TO VARIABLE 'A'
9475:20 F9 EA           JSR MOVFM    ; MOVE ITS CONTENTS INTO FAC
9478:20 BE DE           JSR CHKCOM   ; REMOVE SECOND COMMA
947B:20 E3 DF           JSR PTRGET   ; GET POINTER TO VARIABLE 'B'
947E:20 A7 E7           JSR FSUB     ; FAC = B - A
9481:20 BE DE           JSR CHKCOM   ; REMOVE THIRD & LAST COMMA
9484:20 E3 DF           JSR PTRGET   ; GET POINTER TO VARIABLE 'C'
                        TAX          ; TRANSFER ACC TO X-REG.
9487:AA
9488:20 EB      MULT    JSR MOVMF    ; C = FAC = B - A
948B:60                 RTS          ; RETURN TO BASIC
```

to the fairgoers when they passed by our booth, but you were effective: we can count 25 new members alone from APPLEFEST! We hope you will join us next year. We would also like to give additional thanks to Raines for supplying the booth with the ALF music synthesizer (an unbelievable hit with the crowd, especially when the William Tell Overture came on), Joe for supplying us with the D-TACH 68000 board (excellent graphics, look for his talk in August), and Alan for supplying us with the LOGO demonstration (we easily had more parents and teachers coming to our booth than last year). Again, thank you all!

Several other companies/people are also due our appreciation. We would like to thank Bob Berkowitz at Northeast Expositions for helping us acquire the booth. We would also like to thank the folks at TG Products for the joystick and trackball they donated to the club. These two goodies are currently being reviewed by some of the members of this club, and will be given away as a door prize later this summer. Finally, we would like to thank Dave Szetela for giving us a copy of Mirco-SPARC's THE ASSEMBLER and MACROSOFT. A review of these two software packages will appear in July's NEAT NOTES.

In last year's review, El and I noted many companies simply unveiled improvements to already popular products. This year, however, was witness to many **new** products for our little machines.

Several companies introduced new processor boards for the Apple. There are currently three companies producing processor

```
9410:A9 94      46 START  LDA #<ADD      ; GET THE HIGH BYTE TO
9412:           47 ; THE 'ADD' ROUTINE
9412:A0 52      48        LDY #>ADD      ; GET THE LOW BYTE
9414:20 F2 E2   49        JSR GIVAYF     ; CONVERT THE INTEGER IN
9417:           50 ; A,Y AND STORE IN FAC
9417:20 E3 DF   51        JSR PTRGET     ; GET POINTER TO FIRST
941A:           52 ; PARAM. IN THE AMPERSAND STATEMENT
941A:AA         53        TAX            ; ACC -> X-REG.
941B:20 2B EB   54        JSR MOVMF      ; MOVE FAC TO (Y,X)
941E:20 BE DE   55        JSR CHKCOM     ; REMOVE NEXT COMMA IN LIST
9421:A9 94      56        LDA #<SUB      ; REPEAT SAME STEPS FOR THE
9423:           57 ; 'SUBTRACT' ROUTINE
9423:A0 6F      58        LDY #>SUB
9425:20 F2 E2   59        JSR GIVAYF
9428:20 E3 DF   60        JSR PTRGET
942B:AA         61        TAX
942C:20 2B EB   62        JSR MOVMF
942F:20 BE DE   63        JSR CHKCOM
948C:20 BE DE  106 *MULT  JSR CHKCOM     ; REMOVE FIRST COMMA
948F:20 E3 DF  107        JSR PTRGET     ; GET POINTER TO VARIABLE 'A'
9492:20 F9 EA  108        JSR MOVFM      ; MOVE ITS CONTENTS INTO FAC
9495:20 BE DE  109        JSR CHKCOM     ; REMOVE SECOND COMMA
9498:20 E3 DF  110        JSR PTRGET     ; GET POINTER TO VARIABLE 'B'
949B:20 7F E9  111        JSR FMULT      ; FAC = A * B
949E:20 BE DE  112        JSR CHKCOM     ; REMOVE THIRD & LAST COMMA
94A1:20 E3 DF  113        JSR PTRGET     ; GET POINTER TO VARIABLE 'C'
94A4:AA        114        TAX            ; TRANSFER ACC TO X-REG.
94A5:20 2B EB  115        JSR MOVMF      ; C = FAC = A * B
94A8:60        116        RTS            ; RETURN TO BASIC
94A9:20 BE DE  117 DIV    JSR CHKCOM     ; REMOVE FIRST COMMA
94AC:20 E3 DF  118        JSR PTRGET     ; GET POINTER TO VARIABLE 'A'
94AF:20 F9 EA  119        JSR MOVFM      ; MOVE ITS CONTENTS INTO FAC
94B2:20 BE DE  120        JSR CHKCOM     ; REMOVE SECOND COMMA
94B5:20 E3 DF  121        JSR PTRGET     ; GET POINTER TO VARIABLE 'B'
94B8:20 66 EA  122        JSR FDIV       ; FAC = B / A
94BB:20 BE DE  123        JSR CHKCOM     ; REMOVE THIRD & LAST COMMA
94BE:20 E3 DF  124        JSR PTRGET     ; GET POINTER TO VARIABLE 'C'
94C1:AA        125        TAX            ; TRANSFER ACC TO X-REG.
94C2:20 2B EB  126        JSR MOVMF      ; C = FAC = B / A
94C5:60        127        RTS            ; RETURN TO BASIC

*** SUCCESSFUL ASSEMBLY: NO ERRORS
```

boards based on the advanced Mottorola 68000. Two of the companies, Analytical Engines and PDQ //, were present and demonstrated products they claim use the UCSD p-System (giving you options to run FORTRAN, BASIC, and Pascal). Both of these boards also support Applesoft and DOS 3.3. By all appearances, these boards should be able to run your standard Applesoft programs approximately 5-20 times faster than your current Apple.

One of the most exciting demonstrations was found at the Number Nine booth: these folks demonstrated a new graphics board which is capable of giving horizontal and vertical resolution equal to 1024 pixels by 1024 pixels! This resolution is almost 20 times finer than the resolution you can get from your standard Apple! This board uses the new NEC 7220 graphics processor chip and comes with memory built into the board which is used by the 7220. The 7220 does not tie up the Apple's 6502 processor and is therefore capable of some impressive animation.

T-I-M-E-C-O-R introduced THE OPERATOR, a new modem card for the Apple (see their ad on the inside front cover). Their board connects directly to a standard RJ-11 phone jack, and can be placed in any slot in the Apple. The modem will operate at up to 300 baud (approximately 5 words per second), can be used with both Rotory and Touch Tone phones, and can automatically answer incoming calls. At present, THE OPERATOR will work with ASCII Express Pro, Data Capture, Modem Magic II, TermExec, Transcend I, II, and III, VisiTERM, and Z-Term.

Next month, we will review some of the software introduced at APPLEFEST.

DISK OF THE MONTH

This month's disk of the month comes to us courtesy of the International Apple Corp and L.O.G.I.C, The Loyal Ontario Group Interested in Computers. This disk contains several **excellent** machine language utilities for the Apple. The first set of utilities is B.E.U. (Binary Entry Utility) by Ken Lind. Many machine code listings which appear in our favorite computer magazines are simply printed in their hexadecimal format (the old 8 bytes per line) and are difficult to enter, much less edit. B.E.U. gives you a nice array of commands for entering, editing, and saving machine programs using that particular format.

The second program is MASM, an editor/assembler developed by Scott Jones. Consider getting this disk if you do not already own an assembler: you will have an easier time copying, say, the listings printed in our own **MACHINE LANGUAGE SPOKEN HERE** column.

This disk also contains a handy little disassembler, a version of the old MINI ASSEMBLER for Apples with Applesoft, and an informative machine language tutorial.

If you did not pick up a copy of this disk at our last meeting (June 15), you can acquire a copy by sending $6.50 to Doug McMullin.

MEMBERSHIP APPLICATION

1. Name _____ Date _____
2. Street _____
3. City _____ State _____ Zip _____
4. Phones (a) Home _____ (b) Work _____
5. Software background _____
6. Hardware background _____
7. What will you use your APPLE for? _____
8. In what area can the club assist you? _____
9. Are you available to assist other members? _____
10. Total amount enclosed _____

1983 Membership Dues are

UNITED STATES..... $ 15.00

CANADA..... $ 21.00

(OFFICE USE ONLY)

PAID BY: cash check RECEIPT #_____ STATUS: New Renewal MEM #_____

-DISKETTE ORDER FORM-

Diskettes are sold to members and non-members and can be picked up at the meeting or mailed anywhere in the continental U.S. Circle desired selection.

Games	I	II	III	IV	V	VI	VII					
Education	I	II	Math		I	Scientific			I	Applesoft Tutor		
Pascal	I	II	III	IV	V	VI						
D.O.M.1980					M	J	J	A	S	O	N	D
D.O.M.1981	J	F	M	A	M	J	J	A	S	O	N	D
D.O.M.1982	J	F	M	A	M	J	J	A	S	O	N	D

All D.O.M.'s up to the current month $5.00 picked up, $6.50 mailed. The current D.O.M. is copied free on your initialized diskette at the meeting.

All catagorized diskettes include a description of the programs and are available for $5.00 if picked up and for $6.50 if they must be mailed.

Non-members prices are $10.00 picked up and $11.50 mailed.

Check here if you want diskettes mailed ( ).

NAME _____

ADDRESS _____

CITY/STATE _____

ZIP _____

MEMBERSHIP CARD # _____

Make check payable to N.E.A.T.
Send to:
N.E.A.T.-Attn: Librarian
P.O.Box 2652
Woburn, MA 01801

Total enclosed $ _____.____

neat
New England Apple Tree
Box 2652, Woburn, MA 01801

BULK RATE
U.S. POSTAGE
PAID
NEWTONVILLE, MA. 02160
PERMIT NO. 57931

EXPIRES 02/84

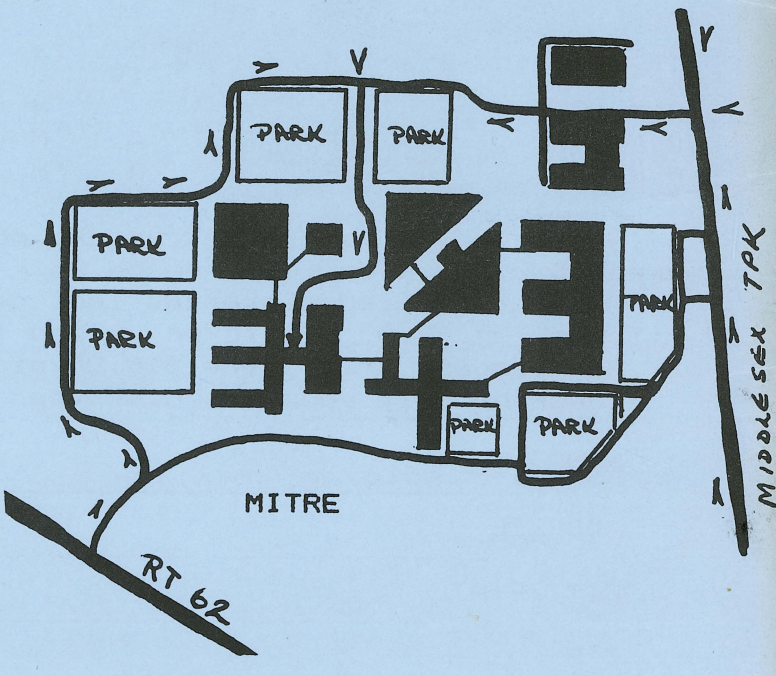
ADVERTISING RATES

Single Issue
- Full Page.....$40.00
- Half Page.....$20.00
- Quarter Page.....$10.00

3 Month series or Longer
- Full Page.....$30.00
- Half Page.....$15.00
- Quarter Page.....$7.50

Inserts letter size accepted at the same rates, subject to weight limitations.

Personal advertisements of members, e.g., swap notices, personal articles for sale, etc. accepted without charge.

Please send ad copy in black and white (half-tones are permited but no bleeds) by the last Wednesday of the month preceeding the month of issue to:

N.E.A.T. Notes
Attention: Editor
P.O.Box 2652
Woburn, MA. 01801